[AIR-Runtime Module Specification]

Purpose:
- Acts as the execution engine of AIR-based DSP pipeline
- Manages stage-wise dataflow, adaptive format switching, and timing

Note: This module was co-developed via GPT-4o-based prompt-driven vibe coding in collaboration with 제현.

---

[1] Module Functions
- init_air_runtime(config): Initialize runtime parameters (latency target, format preferences)
- process_frame(input_pcm): Run full AIR pipeline on single audio frame
- switch_format(stage, new_format): Dynamically change format (e.g., Q31 → FP16)
- set_latency_budget(ms): Set target latency per frame
- get_runtime_stats(): Return processing time per stage, format usage, dropped frames

---

[2] Internal State & Buffers
- PCM_Buffer (input)
- Stage_Buffers: per-stage float/Q31/BFP16 buffers
- Timing_Queue: for stage-wise latency profiling
- Format_Map: dictionary for adaptive switching logic

---

[3] Execution Flow (per frame)
1. PCM input → PCM_Buffer
2. Preprocessing (AGC, NS) → FP16
3. MDCT → BFP16 or Q31
4. Quantization → SPIR-V backend or AIR_QUANT
5. Entropy Coding → int-only block
6. Packet Assembly → AIR_PACK
7. Output frame return + runtime logging

---

[4] Notes
- All switching decisions use latency + quality scoring
- Runtime may cache fast paths (e.g., Q31-only shortcut mode)
- Compatible with AIR-SPIR and AIR-RISCV modules

Author: 제현
Module: AIR-Runtime
Date: 2025-10-15